

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра вычислительных машин, систем и сетей
Дисциплина: Арифметические и логические основы
цифровых устройств

К ЗАЩИТЕ ДОПУСТИТЬ
_____ Ю. А. Луцик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
на тему

ПРОЕКТИРОВАНИЕ И ЛОГИЧЕСКИЙ СИНТЕЗ СУММАТОРА-
УМНОЖИТЕЛЯ ДВОИЧНО-ЧЕТВЕРИЧНЫХ ЧИСЕЛ

БГУИР КР 1-40 02 01 312 ПЗ

Студент
Руководитель

Е.О. Лукьянов
Ю.А. Луцик

МИНСК 2023

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин
Дисциплина: Арифметические и логические основы
цифровых устройств

УТВЕРЖДАЮ
Заведующий кафедрой ЭВМ
_____ Б. В. Никульшин
« ____ » _____ 20 __ г.

ЗАДАНИЕ
по курсовой работе студента
Лукьянова Евгения Олеговича

1. Тема работы: «Проектирование и логический синтез сумматора-умножителя двоично-десятичных чисел»
2. Срок сдачи студентом законченной работы: до 20 мая 2023г.
3. Исходные данные к работе:
 - 3.1. Исходные сомножители: $M_n = 52,33$; $M_t = 37,62$;
 - 3.2. Алгоритм умножения: А;
 - 3.3. Метод умножения: умножение закодированного двоично-четверичного множимого на два разряда двоичного множителя одновременно в прямых кодах;
 - 3.4. Коды четверичных цифр множимого для перехода к двоично-четверичной системе кодирования: $0_4 - 01$, $1_4 - 10$, $2_4 - 00$, $3_4 - 11$;
 - 3.5. Тип синтезируемого умножителя: 2;
 - 3.6. Логический базис для реализации ОЧС: И, НЕ Сумма по модулю; метод минимизации – алгоритм Рота.
 - 3.7. Логический базис для реализации ОЧУС: И, Исключающее ИЛИ, Генератор единицы; метод минимизации – карты Карно-Вейча
4. Содержание пояснительной записки (перечень подлежащих разработке вопросов):

Введение. 1. Разработка алгоритма умножения. 2. Разработка структурной схемы сумматора-умножителя. 3. Разработка функциональных схем основных узлов сумматора-умножителя. 4. Синтез комбинационных схем устройств на основе мультиплексоров. 5. Оценка результатов разработки. Заключение. Список литературы.
5. Перечень графического материала:
 - 5.1. Умножитель-сумматор 2 типа. Схема электрическая структурная.

- 5.2. Однозарядный четверичный сумматор. Схема электрическая функциональная.
- 5.3. Однозарядный четверичный умножитель. Схема электрическая функциональная. Однозарядный четверичный умножитель. Схема электрическая функциональная.
- 5.4. Однозарядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная.
- 5.5. Преобразователь множителя. Схема электрическая функциональная.

КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов курсовой работы	Объём этапа, %	Срок выполнения этапа	Примечания
Разработка алгоритма умножения	10	10.02-20.02	
Разработка структурной схемы сумматора-умножителя	10	21.02-09.03	С выполнением чертежа
Разработка функциональных схем основных узлов сумматора-умножителя	50	10.03-30.04	С выполнением чертежей
Синтез комбинационных схем устройств на основе мультиплексоров	10	01.05-15.05	С выполнением чертежа
Завершение оформления пояснительной записки	20	15.05-20.05	

Дата выдачи задания: 14 февраля 2023г.

Руководитель

_____Ю.А. Луцик

ЗАДАНИЕ ПРИНЯЛ К ИСПОЛНЕНИЮ

_____Е.О. Лукьянов

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	
1. РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ	
2. РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ	
3. РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ	
3.1. Логический синтез одноразрядного четверичного сумматора.....	
3.2. Логический синтез одноразрядного четверичного сумматора-умножителя	
3.3. Логический синтез преобразователя множителя	
4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ	
5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ	
ЗАКЛЮЧЕНИЕ	
СПИСОК ЛИТЕРАТУРЫ	
ПРИЛОЖЕНИЕ А Сумматор-умножитель первого типа. Схема электрическая структурная	
ПРИЛОЖЕНИЕ Б Одноразрядный четверичный сумматор-умножитель. Схема электрическая функциональная.....	
ПРИЛОЖЕНИЕ В Одноразрядный четверичный сумматор. Схема электрическая функциональная.....	
ПРИЛОЖЕНИЕ Г Одноразрядный четверичный сумматор. Реализация на мультиплексорах. Схема электрическая функциональная	
ПРИЛОЖЕНИЕ Д Преобразователь множителя. Схема электрическая функциональная	
ПРИЛОЖЕНИЕ Е Ведомость документов	

ВВЕДЕНИЕ

Курсовое проектирование является обязательным элементом подготовки специалиста с высшим образованием и одной из форм текущей аттестации студента по учебной дисциплине. Для студентов это первая работа такого рода и объёма. Она содержит результаты теоретических и экспериментальных исследований по дисциплине “Арифметические и логические основы вычислительной техники”, включает совокупность аналитических, расчётных, экспериментальных заданий и предполагает выполнение конструкторских работ и разработку графической документации.

Целью данной курсовой работы является проектирование такого цифрового устройства, как двоично-четверичный сумматор-умножитель (СУ). Сумматор является одним из центральных узлов арифметико-логического устройства (АЛУ) вычислительной машины, поэтому глубокое понимание принципов его работы критически важно для современного инженера. Для того чтобы спроектировать данное устройство, необходимо пройти несколько последовательных этапов разработки:

- Разработка алгоритма умножения чисел, по которому работает СУ
- Разработка структурной схемы СУ
- Разработка функциональной схемы основных узлов структурной схемы СУ
- Оценка результатов проделанной работы
- Оформление документации по проделанной работе

В ходе выполнения курсовой работы автором были пройдены все эти этапы. В настоящей пояснительной записке изложено краткое описание процесса проектирования и приведена разработанная автором графическая документация по структурной схеме и функциональным схемам основных её узлов.

1 РАЗРАБОТКА АЛГОРИТМА УМНОЖЕНИЯ

1. Перевод сомножителей из десятичной системы счисления в четверичную.

$$\begin{array}{r} \begin{array}{|l} 52 \\ \hline 52 \\ \hline 0 \end{array} \begin{array}{|l} 4 \\ \hline 13 \\ \hline 12 \end{array} \begin{array}{|l} 4 \\ \hline 3 \end{array} \\ \hline \end{array}$$

$$\begin{array}{r} 0,33 \\ * \quad 4 \\ \hline 1,32 \\ * \quad 4 \\ \hline 1,28 \\ * \quad 4 \\ \hline 1,12 \\ * \quad 4 \\ \hline 0,48 \end{array}$$

$$M_{H4} = 310,1110.$$

В соответствии с заданной кодировкой множимого:

$$M_{H2/4} = 111001,10101001$$

$$\begin{array}{r} \begin{array}{|l} 37 \\ \hline 36 \\ \hline 1 \end{array} \begin{array}{|l} 4 \\ \hline 9 \\ \hline 8 \end{array} \begin{array}{|l} 4 \\ \hline 2 \end{array} \\ \hline \end{array}$$

$$\begin{array}{r} 0,62 \\ * \quad 4 \\ \hline 2,48 \\ * \quad 4 \\ \hline 1,92 \\ * \quad 4 \\ \hline 3,68 \\ * \quad 4 \\ \hline 2,72 \end{array}$$

$$M_{T4} = 211,2132.$$

В соответствии с заданной кодировкой множителя:

$$M_{T2/4} = 100101,10011110$$

2. Запишем сомножители в форме с плавающей запятой в прямом коде:

$$M_H = 0,11100110101001 \quad P_{MH} = 0.0111 + 03_{10}$$

$$M_T = 0,10010110011110 \quad P_{MT} = 0.0011 + 03_{10}$$

Порядок произведения:

$$P_{MH} = 0.0111 \quad 03_4$$

$$\begin{aligned} P_{M_T} &= \frac{0.0011 \ 03_4}{0.1000 \ 12_4} \\ P_{M_H \cdot M_T} &= \end{aligned}$$

Знак произведения определяется суммой по модулю два знаков сомножителей:

$$\text{зн } M_H \oplus \text{зн } M_T = 0 \oplus 0 = 0.$$

При умножении чисел в прямых кодах диада $11(3_4)$ заменяется на триаду $10\bar{1}$. Преобразованный множитель имеет вид $M_T^H = 21122\bar{1}2$. Перемножение мантисс по алгоритму «А» представлено в таблице 1.1

Таблица 1.1 — Перемножение мантисс

Четверичная с/с			Двоично-четверичная с/с			Комментарии
1			2			3
0.	00000000		0.	0101010101010101		\sum_0^q
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_1^q = M_H * 2$
0.	12202220		0.	1000000100000001		\sum_1^q
0.	01220222	0	0.	0110000001000000	01	$\sum_1^q * 4^{-1}$
3.	<u>30232230</u>		1.	<u>1101001100001101</u>		$\Pi_2^q = -1 * M_H$
3.	32113112	0	1.	1100101011101000	01	\sum_2^q
3.	33211311	20	1.	1111001010111010	0001	$\sum_2^q * 4^{-1}$
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_3^q = M_H * 2$
0.	12020131	20	0.	1000010001101110	0001	\sum_3^q
0.	01202013	120	0.	0110000100011011	100001	$\sum_3^q * 4^{-1}$
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_4^q = M_H * 2$
0.	20010233	120	0.	0001011001001111	100001	\sum_4^q
0.	02001023	3120	0.	0100010110010011	11100001	$\sum_4^q * 4^{-1}$
0.	<u>03101110</u>		0.	<u>0111100110101001</u>		$\Pi_5^q = M_H * 1$
0.	11102133	3120	0.	1010100100101111	11100001	\sum_5^q
0.	01110213	33120	0.	0110101001001011	1111100001	$\sum_5^q * 4^{-1}$
0.	<u>03101110</u>		0.	<u>0111100110101001</u>		$\Pi_6^q = M_H * 1$
0.	10211323	33120	0.	1001001010110011	1111100001	\sum_6^q
0.	01021132	333120	0.	0110010010101100	111111100001	$\sum_6^q * 4^{-1}$
0.	<u>12202220</u>		0.	<u>1000000100000001</u>		$\Pi_7^q = M_H * 2$
0.	13230012	333120	0.	1011001101011000	111111100001	\sum_7^q
0.	01323001	2333120	0.	0110110011010110	00111111100001	$\sum_7^q * 4^{-1}$

После окончания умножения необходимо оценить погрешность вычислений. Для этого полученное произведение ($M_{H_4} \cdot M_{T_4} = 0,013230012 \ 333120$, $P_{M_H \cdot M_T} = 7$) приводится к нулевому порядку, а затем переводится в десятичную систему счисления:

$$M_{H4} \cdot M_{T4} = 132300,12333120 \quad P_{M_H} \cdot M_T = 0;$$

$$M_{H10} \cdot M_{T10} = 1968,4368.$$

Результат прямого перемножения операндов дает следующее:
 $M_{H10} \cdot M_{T10} = 91,74 \cdot 17,41 = 1968,6546.$

Абсолютная погрешность:
 $\Delta = 1968,6546 - 1968,4368 = 0,2178.$

Относительная погрешность:

$$\delta = \frac{\Delta}{M_H \cdot M_T} = \frac{0,2178}{1968,4368} = 0,00011 \quad (\delta = 0,011\%)$$

Эта погрешность получена за счёт приближённого перевода из десятичной системы счисления в четверичную обоих сомножителей, а также за счёт округления полученного результата произведения.

2 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ СУММАТОРА-УМНОЖИТЕЛЯ

Если устройство работает как сумматор, то оба слагаемых последовательно (за два такта) заносятся в регистр множимого, а на управляющий вход формирователя дополнительного кода F_2 поступает «1». Необходимо обеспечить выполнение алгоритма сложения чисел, представленных в форме с плавающей запятой, базируясь на схеме умножителя, реализующего заданный алгоритм умножения.

Первое слагаемое переписывается в регистр результата под действием управляющих сигналов, поступающих на входы h всех ОЧУС (рисунок 2.1).

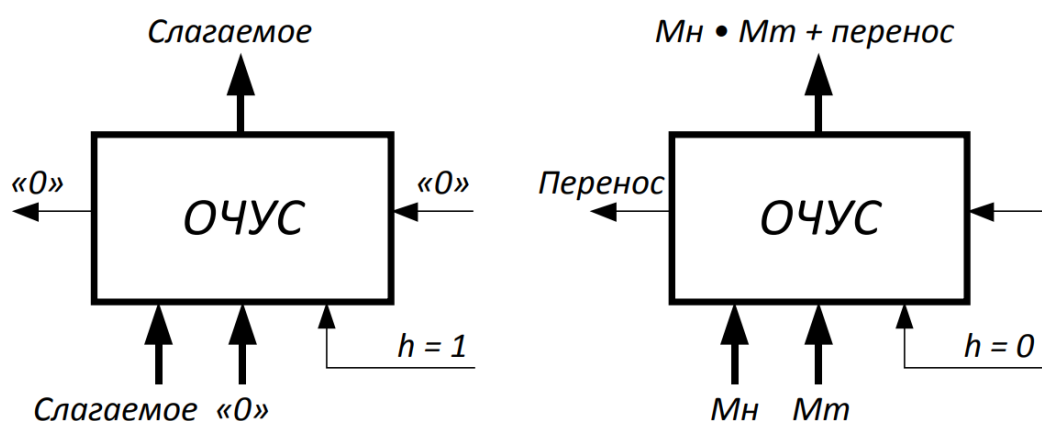


Рисунок 2.1 – Режимы работы ОЧУС

Если на вход h поступает «0», то ОЧУС перемножает разряды M_n и M_t и добавляет к полученному результату перенос из предыдущего ОЧУС.

В ОЧУС первое слагаемое складывается с нулём, записанным в регистре результата, и переписывается без изменений в регистр результата.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУС попадает на входы ОЧУС и складывается с первым слагаемым, хранящимся в регистре результата.

Сумма хранится в регистре результата. Разрядность регистра результата должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения при суммировании переноса.

Если устройство работает как умножитель, то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК F_2 поступает «0».

Диада множителя поступает на входы преобразователя множителя. Единица переноса в следующую диаду, если она возникает, должна быть добавлена к следующей диаде множителя (выход 1 ПМ) в следующем такте, т. е. должна храниться на триггере до следующего такта.

В регистре множителя после каждого такта умножения содержимое сдвигается на два двоичных разряда, и в конце умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения при умножении по алгоритму «А».

Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание ($\overline{01}$). В этом случае инициализируется управляющий вход F_1 формирователя дополнительного кода, и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на -1).

Принцип работы ФДК в зависимости от управляющих сигналов отражён в таблице 2.1.

На выходах 3 и 4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУС вместе с диадами множимого. На трёх выходах ОЧУС формируется результат умножения диад $M_n \cdot M_t$ плюс перенос из предыдущего ОЧУС. Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому перенос, формируемый ОЧУС, может быть только двоичным («0» или «1»):

$$\begin{array}{ccccccc} 3 & \cdot & 2 & = & 12 & (+1 \text{ в случае переноса из} \\ & & & & & \text{предыдущего ОЧУС}) \\ \max & & \max & & \max & \\ M_n & & M_t & & \text{Перенос} \end{array}$$

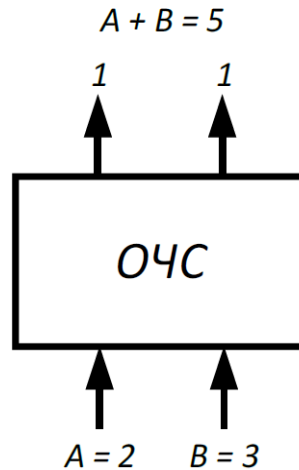
Так как на входы ОЧУС из регистра M_t не могут поступить коды «3», в таблице истинности работы ОЧУС будут содержаться 16 безразличных входных наборов.

Частичные произведения, получаемые на выходах ОЧУС, складываются с накапливаемой частичной суммой из регистра результата с помощью цепочки ОЧС (на первом такте выполняется сложение с нулём).

Частичные суммы хранятся в регистре результата и регистре множителя, т. к. алгоритм умножения «А» предполагает возможность синхронного сдвига этих регистров. Количество тактов умножения определяется разрядностью M_t .

Одноразрядный четверичный сумматор предназначен для сложения двух двоично-четверичных цифр, подаваемых на его входы (рисунок 2.2).

Рисунок 2.2 – Одноразрядный четверичный сумматор



В ОЧС первое слагаемое складывается с нулём, т.к. на старших выходах ОЧУ будут формироваться только коды нуля. Затем первое слагаемое попадает в регистр-аккумулятор, который изначально обнулён.

На втором такте второе слагаемое из регистра множимого через цепочку ОЧУ и ОЧС попадает в аккумулятор, где складывается с первым слагаемым. Таким образом, аккумулятор (накапливающий сумматор) складывает операнды и хранит результат.

Разрядность аккумулятора должна быть на единицу больше, чем разрядность исходных слагаемых, чтобы предусмотреть возможность возникновения переноса при суммировании.

Если устройство работает как умножитель (на входе *Mul/sum* - «0»), то множимое и множитель помещаются в соответствующие регистры, а на управляющий вход ФДК F_2 поступает «0».

Диада множителя поступает на входы преобразователя множителя (ПМ). Задачей ПМ является преобразование диады множителя в соответствии с алгоритмом преобразования. При этом в случае образования единицы переноса в старшую диаду множителя она должна быть учтена при преобразовании следующей старшей диады (выход 1 ПМ), т.е. сохраняться до следующего такта на триггер.

В регистре множителя в конце каждого такта умножения содержимое сдвигается на два двоичных разряда и в последнем такте умножения регистр обнуляется. Это позволяет использовать регистр множителя для хранения младших разрядов произведения при умножении по алгоритму «А» (регистр множителя служит как бы «продолжением» регистра результата).

Выход 2 ПМ переходит в единичное состояние, если текущая диада содержит отрицание (01). В этом случае инициализируется управляющий вход F_1 формирователя дополнительного кода (ФДК) и на выходах ФДК формируется дополнительный код множимого с обратным знаком (умножение на «-1»).

Принцип работы ФДК, в зависимости от управляющих сигналов, приведён в таблице 2.1.

Таблица 2.1 – Режимы работы формирователя дополнительного кода

Сигналы на входах ФДК		Результат на выходах ФДК
F_1	F_2	
0	0	Дополнительный код множимого
0	1	Дополнительный код слагаемого
1	0	Меняется знак Мн
1	1	Меняется знак слагаемого

На выходах 3 и 4 ПМ формируются диады преобразованного множителя, которые поступают на входы ОЧУ вместе с диадами множимого.

ОЧУ предназначен лишь для умножения двух четверичных цифр. Если в процессе умножения возникает перенос в следующий разряд, необходимо предусмотреть возможность его прибавления.

Для суммирования результата умножения текущей диады Мн·Мт с переносом из предыдущей диады предназначен ОЧС. Следовательно, чтобы полностью сформировать частичное произведение четверичных сомножителей, необходима комбинация цепочек ОЧУ и ОЧС.

Частичные суммы формируются в аккумуляторе. На первом этапе он обнулён и первая частичная сумма получается за счёт сложения первого частичного произведения (сформированного на выходах ОЧС) и нулевой частичной суммы (хранящейся в аккумуляторе).

В аккумуляторе происходит сложение i -й частичной суммы с $(i+1)$ -м частичным произведением, результат сложения сохраняется. Содержимое аккумулятора сдвигается на один четверичный разряд вправо в конце каждого такта умножения по алгоритму «А».

На четырёх выходах ОЧУ формируется результат умножения диад Мн·Мт. Максимальной цифрой в диаде преобразованного множителя является двойка, поэтому в старшем разряде произведения максимальной цифрой может оказаться только «1»:

$$\begin{array}{c} 3 \\ \text{max} \\ \text{Мн} \end{array} \cdot \begin{array}{c} 2 \\ \text{max} \\ \text{Мт} \end{array} = \begin{array}{c} 1 \ 2 \end{array}$$

Это означает, что на младшие входы ОЧС никогда не поступят диады цифр, соответствующие кодам «2» и «3», следовательно, в таблице истинности работы ОЧС будут содержаться 16 безразличных входных наборов.

Частичные суммы хранятся в аккумуляторе и регистре множителя, т.к. алгоритм умножения «А» предполагает возможность синхронного сдвига этих устройств. Количество тактов умножения определяется разрядностью Мт.

3 РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ СХЕМ ОСНОВНЫХ УЗЛОВ СУММАТОРА-УМНОЖИТЕЛЯ

3.1 Логический синтез одноразрядного четверичного сумматора-умножителя

ОЧУС – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда из регистра Мн, 2 разряда из регистра Мт и управляющий вход h) и 3 двоичных выхода.

Принцип работы ОЧУС представлен с помощью таблицы истинности (таблица 3.1.1).

Разряды множимого закодированы: 0 – 01, 1 – 10, 2 – 00, 3 – 11;

Разряды множителя закодированы: 0 – 00, 1 – 01, 2 – 10, 3 – 11;

Управляющий вход h определяет тип операции:

«0» – вывод результата умножения закодированных цифр с добавлением переноса из предыдущего ОЧУС, перенос в следующий ОЧУС.

«1» – вывод без изменения значения разрядов, поступивших из регистра множимого, перенос *из* и *в* ОЧУС равны нулю.

В таблице 3.1.1 выделены безразличные наборы, т.к. на входы ОЧУС из разрядов множителя не может поступить код «11».

Таблица 3.1.1 — Таблица истинности ОЧУС

Пер.	Мн.		Мт.		Упр.	Перенос	Результат		Пример операции
p	x_1	x_2	y_1	y_2	h	P	Q_1	Q_2	в четверичной с/с
1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	1	$2 * 0 + 0 = 00$
0	0	0	0	0	1	0	0	0	Выход – код «02»
0	0	0	0	1	0	0	0	0	$2 * 1 + 0 = 02$
0	0	0	0	1	1	0	0	0	Выход – код «02»
0	0	0	1	0	0	1	0	1	$2 * 2 + 0 = 10$
0	0	0	1	0	1	0	0	0	Выход – код «01»
0	0	0	1	1	0	x	x	x	$1 * 3 + 0 = 03$
0	0	0	1	1	1	x	x	x	Выход – код «01»
0	0	1	0	0	0	0	1	0	$3 * 0 + 0 = 00$
0	0	1	0	0	1	0	0	1	Выход – код «03»
0	0	1	0	1	0	0	0	1	$3 * 1 + 0 = 03$
0	0	1	0	1	1	0	0	1	Выход – код «03»
0	0	1	1	0	0	1	1	1	$3 * 2 + 0 = 12$
0	0	1	1	0	1	0	0	1	Выход – код «03»
0	0	1	1	1	0	x	x	x	$3 * 3 + 0 = 21$
0	0	1	1	1	1	x	x	x	Выход – код «03»
0	1	0	0	0	0	0	1	0	$0 * 0 + 0 = 00$
0	1	0	0	0	1	0	1	0	Выход – код «00»

Продолжение таблицы 3.1.1

0	1	0	0	1	0	0	1	0	$0 * 1 + 0 = 00$
0	1	0	0	1	1	0	1	0	Выход – код «00»
0	1	0	1	0	0	0	1	0	$0 * 2 + 0 = 00$
0	1	0	1	0	1	0	1	0	Выход – код «00»
0	1	0	1	1	0	x	x	x	$0 * 3 + 0 = 00$
0	1	0	1	1	1	x	x	x	Выход – код «00»
0	1	1	0	0	0	0	1	0	$2 * 0 + 0 = 00$
0	1	1	0	0	1	0	1	1	Выход – код «02»
0	1	1	0	1	0	0	1	1	$2 * 1 + 0 = 02$
0	1	1	0	1	1	0	1	1	Выход – код «02»
0	1	1	1	0	0	1	1	0	$2 * 2 + 0 = 10$
0	1	1	1	0	1	0	1	1	Выход – код «02»
0	1	1	1	1	0	x	x	x	$2 * 3 + 0 = 12$
0	1	1	1	1	1	x	x	x	Выход – код «02»
1	0	0	0	0	0	x	x	x	$1 * 0 + 1 = 01$
1	0	0	0	0	1	x	x	x	Выход – код «01»
1	0	0	0	1	0	x	x	x	$1 * 1 + 1 = 02$
1	0	0	0	1	1	x	x	x	Выход – код «01»
1	0	0	1	0	0	0	0	1	$1 * 2 + 1 = 03$
1	0	0	1	0	1	x	x	x	Выход – код «01»
1	0	0	1	1	0	x	x	x	$1 * 3 + 1 = 10$
1	0	0	1	1	1	x	x	x	Выход – код «01»
1	0	1	0	0	0	x	x	x	$3 * 0 + 1 = 01$
1	0	1	0	0	1	x	x	x	Выход – код «03»
1	0	1	0	1	0	x	x	x	$3 * 1 + 1 = 10$
1	0	1	0	1	1	x	x	x	Выход – код «03»
1	0	1	1	0	0	1	0	1	$3 * 2 + 1 = 13$
1	0	1	1	0	1	x	x	x	Выход – код «03»
1	0	1	1	1	0	x	x	x	$3 * 3 + 1 = 22$
1	0	1	1	1	1	x	x	x	Выход – код «03»
1	1	0	0	0	0	x	x	x	$0 * 0 + 1 = 01$
1	1	0	0	0	1	x	x	x	Выход – код «00»
1	1	0	0	1	0	x	x	x	$0 * 1 + 1 = 01$
1	1	0	0	1	1	x	x	x	Выход – код «00»
1	1	0	1	0	0	0	0	0	$0 * 2 + 1 = 01$
1	1	0	1	0	1	x	x	x	Выход – код «00»
1	1	0	1	1	0	x	x	x	$0 * 3 + 1 = 01$
1	1	0	1	1	1	x	x	x	Выход – код «00»
1	1	1	0	0	0	x	x	x	$2 * 0 + 1 = 01$
1	1	1	0	0	1	x	x	x	Выход – код «02»
1	1	1	0	1	0	x	x	x	$2 * 1 + 1 = 03$

Продолжение таблицы 3.1.1

1	1	1	0	1	1	х	х	х	Выход – код «02»
1	1	1	1	0	0	1	0	0	$2 * 2 + 1 = 11$
1	1	1	1	0	1	х	х	х	Выход – код «02»
1	1	1	1	1	0	х	х	х	$2 * 3 + 1 = 13$
1	1	1	1	1	1	х	х	х	Выход – код «02»

Минимизация функции Р:

Минимизацию функции Р проведем с помощью карт Вейча. Для функции Р заполненная карта приведена на рисунке 3.1.1. В рисунках 3.1.1 – 3.1.3 символом «х» отмечены наборы, на которых функция может принимать произвольное значение (безразличные наборы).

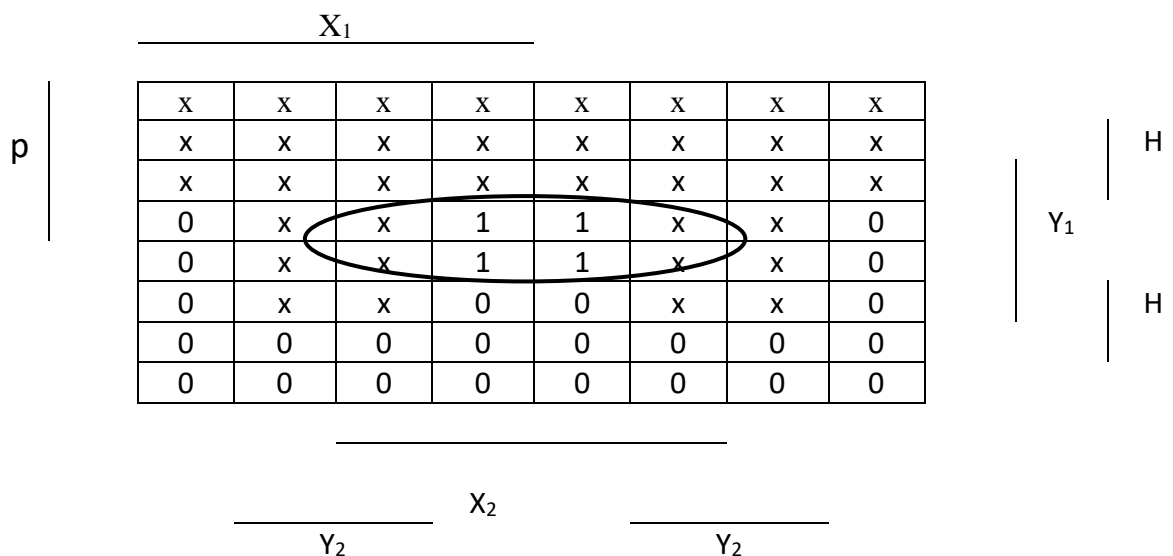


Рисунок 3.1.1 — Минимизация функции Р картой Вейча

Следовательно:

$$P = x_2 y_1 \bar{h}$$

Запишем результат в базисе ИЛИ-НЕ:

$$P = \overline{\bar{x}_2 + \bar{y}_1 + h}$$

Эффективность минимизации можно оценить отношением числа входов схем, реализующих переключательную функцию до и после минимизации:

$$K = \frac{6*4+4+4}{4} = 8$$

Минимизация функции Q_1 :

		X_1									
p		x	x	x	x	x	x	x	x		Y_1
		x	x	x	x	x	x	x	x		
		x	x	x	x	x	x	x	x		
		0	x	x	0	0	x	x	0		
		1	x	x	1	1	x	x	1		
		1	x	x	1	0	x	x	0		
		1	1	1	1	0	0	0	0		
		1	1	1	1	1	0	0	1		
		X_2									
		Y_2				Y_2					

Рисунок 3.1.2 — Минимизация функции Q_1 картой Вейча

Следовательно:

$$Q_1 = \overline{p} \overline{x_2} \overline{h} + \overline{p} x_1$$

Запишем результат в базисе ИЛИ-НЕ:

$$Q_1 = \overline{(p + x_2 + h)} + \overline{(p + \overline{x_1})}$$

Эффективность минимизации:

$$K = \frac{16 \cdot 6 + 16 + 6}{10} = 11,5$$

Минимизация функции Q_2 :

		X_1											
p		x	x	x	x	x	x	x	x		Y ₁		h
		x	x	x	x	x	x	x	x				
		x	x	x	x	x	x	x	x				
		0	x	x	0	1	x	x	1				
		0	x	x	0	1	x	x	1				
		0	x	x	1	1	x	x	0				
		0	0	1	1	1	1	0	0				
		0	0	1	0	0	1	0	0				
		X_2											
		Y_2				Y_2							

Рисунок 3.1.3 — Минимизация функции Q_2 картой Вейча

Следовательно:

$$Q_2 = x_2 y_2 + x_2 h + \bar{x}_1 y_1 \bar{h}$$

Запишем результат в базисе ИЛИ-НЕ:

$$Q_2 = \overline{(\bar{x}_2 + \bar{y}_2)} + \overline{(\bar{x}_2 + \bar{h})} + \overline{(x_1 + \bar{y}_1 + h)}$$

Эффективность минимизации:

$$K = \frac{12 \cdot 6 + 12 + 6}{11} = 8,1$$

3.2 Логический синтез одноразрядного четверичного сумматора

Одноразрядный четверичный сумматор – это комбинационное устройство, имеющее 5 двоичных входов (2 разряда одного слагаемого, 2 разряда второго слагаемого и вход переноса) и 3 двоичных выхода.

Принцип работы ОЧС представлен с помощью таблицы истинности (таблица 3.2.1)

Кодировка слагаемых обоих разрядов: 0 – 10, 1 – 00, 2 – 11, 3 – 01;

Таблица 3.2.1 — Таблица истинности ОЧС

a_1	a_2	b_1	b_2	p	Π	S_1	S_2	<i>Пример операции в четверичной с/с</i>
1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	1	1	$1 + 1 + 0 = 02$
0	0	0	0	1	0	0	1	$1 + 1 + 1 = 03$
0	0	0	1	0	1	1	0	$1 + 3 + 0 = 10$
0	0	0	1	1	1	0	0	$1 + 3 + 1 = 11$
0	0	1	0	0	0	0	0	$1 + 0 + 0 = 01$
0	0	1	0	1	0	1	1	$1 + 0 + 1 = 02$
0	0	1	1	0	0	0	1	$1 + 2 + 0 = 03$
0	0	1	1	1	1	1	0	$1 + 2 + 1 = 10$
0	1	0	0	0	1	1	0	$3 + 1 + 0 = 10$
0	1	0	0	1	1	0	0	$3 + 1 + 1 = 11$
0	1	0	1	0	1	1	1	$3 + 3 + 0 = 12$
0	1	0	1	1	1	0	1	$3 + 3 + 1 = 13$
0	1	1	0	0	0	0	1	$3 + 0 + 0 = 03$
0	1	1	0	1	1	1	0	$3 + 0 + 1 = 10$
0	1	1	1	0	1	0	0	$3 + 2 + 0 = 11$
0	1	1	1	1	1	1	1	$3 + 2 + 1 = 12$
1	0	0	0	0	0	0	0	$0 + 1 + 0 = 01$
1	0	0	0	1	0	1	1	$0 + 1 + 1 = 02$
1	0	0	1	0	0	0	1	$0 + 3 + 0 = 03$

Продолжение таблицы 3.2.1

1	0	0	1	1	1	1	0	$0 + 3 + 1 = 10$
1	0	1	0	0	0	1	0	$0 + 0 + 0 = 00$
1	0	1	0	1	0	0	0	$0 + 0 + 1 = 01$
1	0	1	1	0	0	1	1	$0 + 2 + 0 = 02$
1	0	1	1	1	0	0	1	$0 + 2 + 1 = 03$
1	1	0	0	0	0	0	1	$2 + 1 + 0 = 03$
1	1	0	0	1	1	1	0	$2 + 1 + 1 = 10$
1	1	0	1	0	1	0	0	$2 + 3 + 0 = 11$
1	1	0	1	1	1	1	1	$2 + 3 + 1 = 12$
1	1	1	0	0	0	1	1	$2 + 0 + 0 = 02$
1	1	1	0	1	0	0	1	$2 + 0 + 1 = 03$
1	1	1	1	0	1	1	0	$2 + 2 + 0 = 10$
1	1	1	1	1	1	0	0	$2 + 2 + 1 = 11$

Минимизация функции П:

Определим множество единичных кубов:

$$L = \left\{ \begin{array}{l} 00010, 00011, 00111, \\ 01000, 01001, 01010, \\ 01011, 01101, 01110, \\ 01111, 10011, 11001, \\ 11010, 11011, 11110, \\ 11111 \end{array} \right\}$$

Множество безразличных кубов пустое.

Сформируем множество $C_0 = L \cup N$:

$C_0 = \{00010, 00011, 00111, 01000, 01001, 01010, 01011, 01101, 01110, 01111, 10011, 11001, 11010, 11011, 11110, 11111\}$

Первым этапом алгоритма Рота является нахождение множества простых импликант.

Для реализации этого этапа будем использовать операцию умножения (*) над множествами C_0 , C_1 и т. д., пока в результате операции будут образовываться новые кубы большей размерности.

Первый шаг умножения ($C_0 * C_0$) приведён в таблице 3.2.2.

Таблица 3.2.2 – Поиск простых импликант ($C_0 * C_0$)

$C_0 * C_0$	00010	00011	00111	01000	01001	01010	01011	01101	01110	01111	10011	11001	11010	11011	11110	11111
00010	-															
00011	0001x	-														
00111		00x11	-													
01000				-												
01001				0100x	-											
01010	0x010			010x0		-										
01011		0x011			010x1	0101x	-									
01101					01x01			-								
01110						01x10			-							
01111			0x111				01x11	011x1	0111x	-						
10011		x0011									-					
11001					x1001							-				
11010						x1010							-			
11011							x1011				1x011	110x1	1101x	-		
11110									x1110				11x10		-	
11111										x111				11x11	1111x	-

В результате этой операции сформируется новое множество кубов:

$$C_1 = \{0001x, 0x010, 00x11, 0x011, x0011, 0x111, 0100x, 010x0, 010x1, 01x01, x1001, 0101x, 01x10, x1010, 01x11, x1011, 011x1, 0111x, x1110, x1111, 1x011, 110x1, 1101x, 11x10, 11x11, 1111x\}$$

Множество Z_0 кубов, не участвовавших в образовании новых кубов, пустое.

В таблице 3.2.3 приведён следующий шаг поиска простых импликант с помощью операции $C_1 * C_1$.

Продолжение таблицы 3.2.3

[illegible]

Продолжение таблицы 3.2.3

$C_1 * C_1$	x1010	01x11	x1011	011x1	0111x	x1110	x1111	1x011	110x1	1101x	11x10	11x11	1111x
0001x													
0x010													
00x11													
0x011													
x0011													
0x111													
0100x													
010x0													
010x1													
01x01													
x1001													
0101x													
01x10													
x1010	-												
01x11		-											
x1011	x101x		-										
011x1				-									
0111x					-								
x1110	x1x10					-							
x1111			x1x11			x111x	-						

Продолжение таблицы 3.2.3

1x011								-					
110x1									-				
1101x										-			
11x10											-		
11x11		x1x11									11x1x	-	
1111x					x111x					11x1x			-

В результате образовалось множество C_2 кубов второй размерности:

$$C_2 = \{0x01x, 0xx11, xx011, 010xx, 01xx1, x10x1, 01x1x, x101x, x1x10, x1x11, x111x, 11x1x\}$$

Множество Z_1 кубов, не участвовавших в образовании новых кубов, пустое.

В таблице 3.2.4 приведён следующий шаг поиска простых импликант – операция $C_2 * C_2$

Таблица 3.2.4 – Поиск простых импликант $C_2 * C_2$

$C_2 * C_2$	0x01x	0xx11	xx011	01xx1	01x1x	x111x	010xx	x10x1	x1x11	x101x	x1x10	11x1x
0x01x	-											
0xx11		-										
xx011			-									
01xx1				-								
01x1x					-							
x111x						-						
010xx							-					
x10x1								-				
x1x11									-			
x101x						x1x1x				-		
x1x10									x1x1x		-	
11x1x					x1x1x							-

В результате образовалось множество C_3 кубов третьей размерности:
 $C_3 = \{x1x1x\}$.

Получено множество $Z_2 = \{0x01x, 0xx11, xx011, 010xx, 01xx1, x10x1\}$.

В таблице 3.2.5 приведён следующий шаг поиска простых импликант – операция $C_3 * C_3$.

Таблица 3.2.5 – Поиск простых импликант $C_3 * C_3$

$C_3 * C_3$	$x1x1x$
$x1x1x$	-

Новых кубов (четвертой размерности) не образовалось.
Получено множество $Z_3 = \{x1x1x\}$

На этом заканчивается этап поиска простых импликант.

Множество простых импликант:

$$Z = Z_0 \cup Z_1 \cup Z_2 \cup Z_3 = \{0x01x, 0xx11, xx011, 010xx, 01xx1; x10x1, x1x1x\}$$

Следующий этап – поиск L-экстремалей на множестве простых импликант (таблица 3.2.6). Для этого используется операция # (вычитание).

Таблица 3.2.6 – Поиск L-экстремалей

$z\#(Z-z)$	$0x01x$	$0xx11$	$xx011$	$010xx$	$01xx1$	$x10x1$	$x1x1x$
$0x01x$	-	$zz1zz$ $0x111$	$1zzzz$ $1x011$	$zzz0z$ $0100x$	$zz10z$ $011x1$ $01x01$	$1zz0z$ $110x1$ $x1001$	$1z1zz$ $11x1x$ $x111x$
$0xx11$	$zzzz0$ $0x010$	-	$yzzzz$ $1x011$	$0zzy0$ $0100x$	$zzz0z$ 01101 $zzzyz$ $01x01$	$yzz0z$ $110x1$ $1zzyz$ $x1001$	$yzzz0$ $11x1x$ $1zzz0$ $1111x$ $x1110$

Продолжение таблицы 3.2.6

xx011	zzzzy 0x010	zzyzz 0x111	-	zzzy0 0100x	zzyyz 01101 zzlyz 01x01	zzz0z 11001 zzyyz x1001	zz1z0 1111x 11x10 zzyzy 1111x zzyzy x1110
010xx	z0zzz 00010	z0yzz 0x111	y0zzz 1x011	-	zzyzz 01101 zz1zz 01101	yzzzz 11001 1zzzz 11001	yzyzz 1111x yzlzz 11x10 yzyzz 1111x 1zyzz x1110
01xx1	zyzzy 00010	z0zzz 00111	y0zzz 1x011	zzzz0 01000	-	yzzzz 11001 yzzzz 11001	yzzzz 1111x yzzzy 11x10 yzzz0 1111x 1zzzy x1110
x10x1	zyzzy 00010	zyyzz 00111	z0zzz 10011	zzzzy 01000	zzyzz 01101 zzyzz 01101	-	zzyz0 1111x zzlzy 11x10 zzyz0 1111x zzyzy x1110
x1x1x	zyzzz 00010	zyzzz 00111	zyzzz 10011	zzzyz 01000	zzzyz 01101 zzzyz 01101	zzzyz 11001 zzzyz 11001	-
Остаток	00010	00111	10011	01000	01101 01101	11001 11001	1111x 11x10 1111x x1110

В таблице 3.2.6 из каждой простой импликанты поочерёдно вычитаются все остальные простые импликанты $Z \setminus (Z - z)$.

Множество L-экстремалей = $E = \{0x01x; 0xx11; xx011; 010xx; 01xx1; x10x1; x1x1x\}$

Исходные кубы не надо анализировать так как все они покрываются найденной L-экстремалью. Поиск минимального покрытия завершён.

$$\Pi = \bar{a}_1 a_2 p + \bar{a}_1 a_2 \bar{b}_1 + \bar{a}_1 b_2 p + \bar{b}_1 b_2 p + \bar{a}_1 \bar{b}_1 b_2 + a_2 \bar{b}_1 p + a_2 b_2$$

Запишем результат в базисе И-Константная единица-Сумма по модулю:

$$\Pi = (((a_1 \cdot (a_2 \oplus 1) \cdot (p \oplus 1)) \oplus 1) \cdot ((a_1 \cdot (a_2 \oplus 1) \cdot (b_1 \oplus 1)) \oplus 1) \cdot ((a_1 \cdot (b_2 \oplus 1) \cdot (p \oplus 1)) \oplus 1) \cdot ((b_1 \cdot (b_2 \oplus 1) \cdot (p \oplus 1)) \oplus 1) \cdot (((a_1 \oplus 1) \cdot b_1 \cdot (b_2 \oplus 1)) \oplus 1) \cdot (((a_2 \oplus 1) \cdot b_1 \cdot (p \oplus 1)) \oplus 1) \cdot (((a_2 \oplus 1) \cdot (b_2 \oplus 1)) \oplus 1)$$

Эффективность минимизации:

$$K = \frac{16 \cdot 5 + 16 + 5}{28} = 3,6$$

Минимизация функции S_1

Минимизацию функции S_1 проведем с помощью карт Карно. Для функции S_1 заполненная карта приведена на рисунке 3.2.7.

$\begin{matrix} b_1 b_2 \Pi \\ a_1 a_2 \end{matrix}$	000	001	011	010	110	111	101	100
00		0	0		0			0
01		0	0		0			0
11	0			0		0	0	
10	0			0		0	0	

Рисунок 3.2.1 — Минимизация функции S_1 картой Карно

$$\text{Следовательно: } S_1 = \bar{a}_1 \bar{b}_1 \bar{p} + a_1 \bar{b}_1 p + \bar{a}_1 b_1 p + a_1 b_1 \bar{p}$$

Запишем результат в базисе И-Константная единица-Сумма по модулю:

$$S_1 = ((a_1 \cdot b_1 \cdot p) \oplus 1) \cdot (((a_1 \oplus 1) \cdot b_1 \cdot (p \oplus 1)) \oplus 1) \cdot ((a_1 \cdot (b_1 \oplus 1) \cdot (p \oplus 1)) \oplus 1) \cdot (((a_1 \oplus 1) \cdot (b_1 \oplus 1) \cdot p) \oplus 1)$$

Эффективность минимизации:

$$K = \frac{10 \cdot 5 + 10 + 5}{21} = 3,1$$

Минимизация функции S_2

Минимизацию функции S_2 проведем с помощью карт Карно. Для функции S_2 заполненная карта приведена на рисунке 3.2.8.

$\begin{matrix} b_1 b_2 \backslash a_1 a_2 \end{matrix}$	000	001	011	010	110	111	101	100
00	1	1	0	0	1	0	1	0
01	0	0	1	1	0	1	0	1
11	1	0	1	0	0	0	1	1
10	0	1	0	1	1	1	0	0

Рисунок 3.2.2 — Минимизация функции S_2 картой Карно

Следовательно:

$$S_2 = \bar{a}_1 \bar{a}_2 \bar{b}_2 p + \bar{a}_2 \bar{b}_1 \bar{b}_2 p + \bar{a}_1 \bar{a}_2 \bar{b}_1 \bar{b}_2 + \bar{a}_2 b_1 b_2 \bar{p} + a_1 \bar{a}_2 b_1 b_2 + a_1 \bar{a}_2 b_2 p + a_1 \bar{a}_2 b_2 \bar{p} + \bar{a}_1 a_2 \bar{b}_1 b_2 + a_2 \bar{b}_1 b_2 p + \bar{a}_1 a_2 b_2 p + a_2 b_1 \bar{b}_2 \bar{p} + a_1 a_2 b_1 \bar{b}_2 + a_1 a_2 \bar{b}_2 \bar{p}$$

Запишем результат в базисе И-Константная единица-Сумма по модулю:

$$S_2 = (((a_1 \cdot a_2 \cdot b_2 \cdot (p \oplus 1)) \oplus 1) \cdot ((a_2 \cdot b_1 \cdot b_2 \cdot (p \oplus 1)) \oplus 1) \cdot ((a_1 \cdot a_2 \cdot b_1 \cdot b_2) \oplus 1) \cdot ((a_2 \cdot (b_1 \oplus 1) \cdot (b_2 \oplus 1) \cdot p) \oplus 1) \cdot (((a_1 \oplus 1) \cdot a_2 \cdot (b_1 \oplus 1) \cdot (b_2 \oplus 1)) \oplus 1) \cdot (((a_1 \oplus 1) \cdot a_2 \cdot (b_2 \oplus 1) \cdot (p \oplus 1)) \oplus 1) \cdot (((a_1 \oplus 1) \cdot a_2 \cdot (b_2 \oplus 1) \cdot p) \oplus 1) \cdot ((a_1 \cdot (a_2 \oplus 1) \cdot b_1 \cdot (b_2 \oplus 1)) \oplus 1) \cdot (((a_2 \oplus 1) \cdot b_1 \cdot (b_2 \oplus 1) \cdot (p \oplus 1)) \oplus 1) \cdot ((a_1 \cdot (a_2 \oplus 1) \cdot (b_2 \oplus 1) \cdot (p \oplus 1)) \oplus 1) \cdot (((a_2 \oplus 1) \cdot (b_1 \oplus 1) \cdot b_2 \cdot p) \oplus 1) \cdot (((a_1 \oplus 1)) \cdot (a_2 \oplus 1) \cdot (b_1 \oplus 1) \cdot b_2) \oplus 1) \cdot (((a_1 \oplus 1) \cdot (a_2 \oplus 1) \cdot b_2 \cdot p) \oplus 1)$$

Эффективность минимизации:

$$K = \frac{16 \cdot 5 + 16 + 5}{92} = 1,1$$

3.3. Логический синтез преобразователя множителя

Преобразователь множителя (ПМ) – это устройство, которое преобразовывает диады множителя в соответствии с методом умножения.

При умножении в дополнительных кодах ПМ заменяет диады 11 (3_4) и 10 (2_4) на триады $1\overline{0}1$ и $1\overline{1}0$ соответственно.

Принцип работы ПМ представлен с помощью таблицы истинности (таблица 3.3.1).

Таблица 3.3.1 – Таблица истинности ПМ

Входная диада		Младший разряд	Знак	Выходная диада	
a_1	a_2	p	Q	S_1	S_2
1	2	3	4	5	6
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	0	0	0

Минимизацию переключательных функций произведём с помощью карт Вейча и реализуем их в базисе И, ИЛИ, НЕ.

Функция Q

Для функции Q заполненная карта приведена на рисунке 3.3.1.

		a_2		
		—		
a_1		1	1	
		1	1	
		—		
		p		

Рисунок 3.3.1 – Минимизация функции Q при помощи карты Вейча

Следовательно:

$$Q = a_1 \bar{a}_2 + a_1 \bar{p} = a_1 (\bar{a}_2 + \bar{p}).$$

Функция S_1

Для функции S_1 Заполненная карта приведена на рисунке 3.3.2.

	a_2			
a_1				1
		1		
	p			

Рисунок 3.3.2 – Минимизация функции S_1 при помощи карты Вейча

Следовательно:

$$S_1 = \bar{a}_1 a_2 p + a_1 \bar{a}_2 \bar{p}.$$

Функция S_2

Для функции S_2 заполненная карта приведена на рисунке 3.3.3.

	a_2			
a_1	1		1	
	1		1	
	p			

Рисунок 3.3.3 – Минимизация функции S_2 при помощи карты Вейча

Следовательно:

$$S_2 = p \oplus a_2.$$

Функциональная схема ПМ приведена в приложении Д.

4. СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ УСТРОЙСТВ НА ОСНОВЕ МУЛЬТИПЛЕКСОРОВ

Мультиплексор – это логическая схема, которая имеет n информационных входов, m управляющих входов и один выход. При этом должно выполняться условие $m = 2^n$.

На выход мультиплексора может быть пропущен без изменений один любой логический сигнал, поступающий на один из информационных входов. Порядковый номер информационного входа, значение которого в данный момент должно быть передано на выход, определяется двоичным кодом, подаваемым на управляющие входы.

Переключательные функции (ПФ) от пяти переменных (как, например, ОЧС) можно реализовать на мультиплексоре «один из восьми».

Управляющее поле такого мультиплексора будет определяться тремя переменными, следовательно, число групп с одинаковыми значениями этих переменных будет равно восьми. Также, реализация нескольких ПФ требует для каждой ПФ отдельного мультиплексора.

Для определения управляющего поля мультиплексора возьмём переменные a_1 , a_2 и b_1 .

Таблица истинности для синтеза ПФ ОЧС приведена в таблице 4.1.

Таблица 4.1 – Таблица истинности для синтеза ПФ ОЧС

a_1	a_2	b_1	b_2	p	Π	Функция	S_1	Функция	S_2	Функция
1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	0	b_2	1	\bar{p}	1	\bar{b}_2
0	0	0	0	1	0		0		1	
0	0	0	1	0	1		1		0	
0	0	0	1	1	1		0		0	
0	0	1	0	0	0	$b_2 \cdot p$	0	p	0	$b_2 \oplus p$
0	0	1	0	1	0		1		1	
0	0	1	1	0	0		0		1	
0	0	1	1	1	1		1		0	
0	1	0	0	0	1	1	1	\bar{p}	0	b_2
0	1	0	0	1	1		0		0	
0	1	0	1	0	1		1		1	
0	1	0	1	1	1		0		1	
0	1	1	0	0	0	$b_2 + p$	0	p	1	$b_2 \oplus p$
0	1	1	0	1	1		1		0	
0	1	1	1	0	1		0		0	
0	1	1	1	1	1		1		1	

Продолжение таблицы 4.1

1	0	0	0	0	0	$b_2 \cdot p$	0	p	0	$b_2 \oplus p$
1	0	0	0	1	0		1		1	
1	0	0	1	0	0		0		1	
1	0	0	1	1	1		1		0	
1	0	1	0	0	0	0	1	\bar{p}	0	b_2
1	0	1	0	1	0		0		0	
1	0	1	1	0	0		1		1	
1	0	1	1	1	0		0		1	
1	1	0	0	0	0	$b_2 + p$	0	p	1	$b_2 \oplus p$
1	1	0	0	1	1		1		0	
1	1	0	1	0	1		0		0	
1	1	0	1	1	1		1		1	
1	1	1	0	0	0	b_2	1	\bar{p}	1	\bar{b}_2
1	1	1	0	1	0		0		1	
1	1	1	1	0	1		1		0	
1	1	1	1	1	1		0		0	

При синтезе ОЧС на основе мультиплексоров получается более эффективная схема (если считать, что количество входов мультиплексора = $n + 2^n$, не включая входы элементов, из которых он состоит).

Функциональная схема ОЧС на основе мультиплексоров представлена в приложении Г.

5. ОЦЕНКА РЕЗУЛЬТАТОВ РАЗРАБОТКИ

Формула расчёта временных затрат на умножение:

$$T = n * (T_{\text{ПМ}} + T_{\text{ФДК}} + m * T_{\text{ОЧУС}} + 3 * T_{\text{ОЧС}} + T_{\text{сдвига}}), \text{ где}$$

$T_{\text{ПМ}}$ – время преобразования множителя;

$T_{\text{ФДК}}$ – время формирования дополнительного кода множимого;

$T_{\text{ОЧУС}}$ – время умножения на ОЧУС;

$T_{\text{ОЧС}}$ – время формирования единицы переноса в ОЧС;

$T_{\text{сдвига}}$ – время сдвига в регистрах;

n – количество разрядов множителя;

m – количество разрядов множимого.

Минимизация функций позволила в несколько раз удешевить схему сумматора-умножителя и уменьшить затраты времени на выполнение за счет уменьшения количества элементов.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы была разработана структурная схема сумматора-умножителя второго типа, а также функциональные схемы основных узлов данного устройства. Для уменьшения стоимости логических схем были выполнены минимизации переключательных функций различными способами. Такой подход позволил выявить достоинства и недостатки этих алгоритмов.

В качестве главного достоинства минимизации картами Карно-Вейча можно выделить простоту и минимальные затраты времени. Однако применение данного способа для функций многих переменных будет затруднительно. Для минимизации функций многих переменных удобно использовать алгоритм Рота, который полностью формализует алгоритмы минимизации и делает минимизацию доступной для выполнения компьютерной программой.

Функциональные схемы были построены в различных логических базисах. Это позволило закрепить теоретические знания основных законов булевой алгебры, например, правило де Моргана.

Реализация переключательных функций на основе мультиплексоров позволила облегчить процесс минимизации этих функций и упростить функциональную схему одноразрядного четверичного сумматора.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Луцик, Ю. А. Учебное пособие по курсу «Арифметические и логические основы вычислительной техники» / Ю. А. Луцик, И. В. Лукьянова. - Минск : БГУиР, 2014. – 76с.

Искра, Н. А. Арифметические и логические основы вычислительной техники : пособие / Н. А. Искра, И. В. Лукьянова, Ю. А. Луцик. – Минск : БГУИР, 2016. – 75 с.

Единая система конструкторской документации (ЕСКД) : справ. пособие / С. С. Борушек [и др.]. – М. : Изд-во стандартов, 1989. – 352 с.

Лысиков, Б. Г. Цифровая вычислительная техника / Б. Г. Лысиков. – Минск : Выш. шк., 2003. – 242 с.

Савельев, А. Я. Прикладная теория цифровых автоматов / А. Я. Савельев. – М. : Высш. шк., 1987. – 272 с.

ПРИЛОЖЕНИЕ А

(обязательное)

Сумматор-умножитель второго типа. Схема электрическая структурная

ПРИЛОЖЕНИЕ Б

(обязательное)

Одноразрядный четверичный сумматор. Схема электрическая функциональная

ПРИЛОЖЕНИЕ В

(обязательное)

Одноразрядный четверичный сумматор-умножитель. Схема
электрическая функциональная

ПРИЛОЖЕНИЕ Г

(обязательное)

Однозарядный четверичный сумматор. Схема электрическая функциональная
на основе мультиплексоров

ПРИЛОЖЕНИЕ Д
(обязательное)

Преобразователь множителя. Схема электрическая функциональная

ПРИЛОЖЕНИЕ Е
(обязательное)
Ведомость документов